Program and Implementation of Stack Data Structure in C++ Using Arrays

Posted by Lalith Rallabhandi at 11:55

Stack is an **Abstract Data Type** and common Data Structure . Stack is an ordered list which is used in performing the the operations such as Insertion (Push) , Deletion (Pop) , Top can be performed easily.
Stack refers to "**Last In First Out**" Principle (LIFO)

Applications of Stack :

As there are many applications in the field of computer science . Here I will list few major applications of the stack .

1. Stack Data structures is mainly used in Evaluating the Expressions and Syntax parsing .
2. Stack Data Type is used in conversion of Decimal number to Binary number.
3. Stack is used in the application of the Quick Sort to sort a given array or List . Quick sort is one of the efficient sorting techniques which is based on Divide and conquer algorithm .
4. BackTracking is other major applications of stack . In the maze problems backtracking helps to trace the previous path and each path is stored in the form of stack data structure .

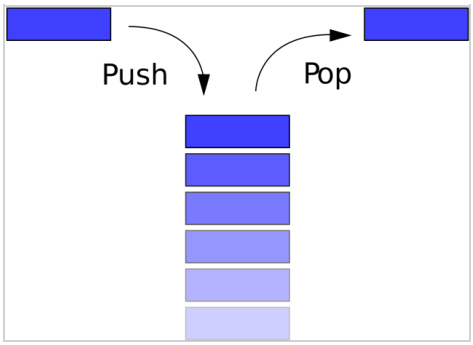


Image credits:my.opera.com

Push :

Push is an operation or function which helps in inserting an element into the stack .

Pop :

Pop is an operation similar to Deletion which helps in deleting or removing an element from the stack .

Top :

Top is a function used to display topmost element in the stack .

Also check : Implementation and program for complete Binary Tree using C++

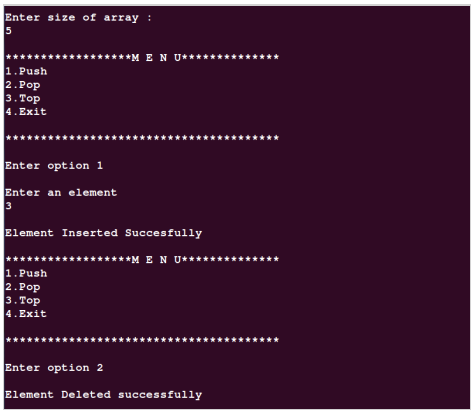
Program for Stack and Implementation Using C++

```
#include<iostream>
using namespace std;
template <class T>
class stack
{
public :
 T *a;
 int top,size;
 stack()
 {
  top=-1;
  cout<<"\nEnter size of array :"<<endl;
  cin>>size;
  a=new T[size];
 }
 int isfull()
 {
 if(top==(size-1))
  return 1;
 else
  return 0;
 }
```

```
 int isempty()
 {
 if(top==-1)
  return 1;
 else
  return 0;
 }
 void topp()
 {
 if(isempty())
  cout<<"\nStack Underflow"<<endl;
 else
  cout<<"\nTop Element is "<<a[top-1]<<endl;
 }
 void push()
 {
 T n;
 if(isfull())
  cout<<"\nStack Overflow"<<endl;
 else
 {
  cout<<"\nEnter an element"<<endl;
  cin>>n;
  a[top++]=n;
  cout<<"\nElement Inserted Succesfully"<<endl;
 }
 }
 void pop()
 {
 if(isempty())
  cout<<"\nStack Underflow"<<endl;
 else
  top=top-1;
  cout<<"\nElement Deleted successfully"<<endl;
 }
};
int main()
{
stack <int>s;
int i=0,k;
while(i!=1)
{
```

```
 cout<<"\n******************M E N U*************\n";
 cout<<"1.Push\n2.Pop\n3.Top\n4.Exit\n";
 cout<<"\n*************************************\n";
 cout<<"\nEnter option ";
 cin>>k;
switch(k)
{
 case 1:
 s.push();
 break;
 case 2:
 s.pop();
 break;
 case 3:
 s.topp();
 break;
 case 4:
 i=1;
 break;
 default :
 cout<<"\n------- Wrong Option -------\n";
 break;
}
}
return 0;
}
```

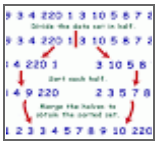
Output of the Program :



The above program is implemented using Templates and Object Oriented Programming in c++ .

If I have missed anything Please get it to me through comments . Also comment your logic and idea for improvement of the program . If you feel the program for stack can be implemented in another best way please comment below .

You Might Also Like:



Merge sort Algorithm and Program in...

Quicksort Program in C & C++ and Al...

Insertion sort Algorithm and Insert...

Algorithm and Program for Depth Fir...

Labels: Algorithm, C++, Data Structures, Programming

0 COMMENTS:

POST A COMMENT

Enter your comment...

Comment as: Google Accoun

Publish Preview

Newer Post Home Older Post